Description

REMOTE MONITORING, CONFIGURING, PROGRAMMING AND DIAGNOSTIC SYSTEM AND METHOD FOR VEHICLES AND VEHICLE COMPONENTS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part claiming the benefit of commonly assigned, co-pending U.S. Patent Application. No. 09/640,785, filed August 18, 2000 entitled "System, Method and Computer Program Product for Remote Vehicle Diagnostics, Monitoring, Configuring and Reprogramming" to Bromley et al., the disclosure of which is incorporated by reference in its entirety. This application also claims the benefit of U.S. Provisional Application No. 60/351,165 (Attorney Docket No. 65855-0060), entitled "Wireless Communication Framework," filed January 23, 2002.

BACKGROUND

TECHNICAL FIELD

[0002] The present invention relates generally to computer data and information systems, and more particularly to computer tools for storing, processing, and displaying vehicle information.

RELATED ART

[0003] It is common for companies to own a large number, or fleet, of commercial motor vehicles. Typical examples of such companies include commercial courier services, moving companies, freight and trucking companies, truck leasing companies, as well as passenger vehicle leasing companies and passenger carriers. To maintain profitability, a company owning a vehicle fleet ideally minimizes the time spent in vehicle maintenance and repair. Maintaining optimum vehicle performance often involves removing vehicles from service to conduct fault analysis, scheduled maintenance, diagnostics monitoring and parameter modifications.

[0004] Further, companies that manufacture vehicle components may wish to have a central database to access information for product improvements, warranty service, diagnostics, and other component data after components have been

installed on the vehicle. Because different companies and different industries have different vehicle data gathering and reporting needs, current solutions involve constructing specialized systems for each particular user application.

[0005]   There is a desire for a system that can monitor, configure, program and diagnose vehicles and/or vehicle components while allowing customization of the vehicle data to accommodate the different needs of different users and different.

SUMMARY

[0006]   Accordingly, one embodiment is directed to a system for monitoring, configuring, programming and/or diagnosing operation of at least one vehicle, comprising an on-board unit disposed on the vehicle to send and receive data corresponding to at least one vehicle operating characteristic, a plurality of modular applications, each application having an associated function that processes the data corresponding to said at least one vehicle operating characteristic obtained via the on-board unit, and an interface that allows selection among the plurality of modular applications to create a customized system.

[0007]   Another embodiment is directed to an on-board unit dis-

posed on a vehicle for use in a system for monitoring, configuring, programming and/or diagnosing operation of at least one vehicle, comprising at least one on-board unit interface to support communication between the on-board unit and at least one device outside the on-board unit, a processor that manages the data sent and received by the on-board unit via said at least one interface, and a memory coupled to the processor.

[0008] A further embodiment is directed to a method for monitoring, configuring, programming and/or diagnosing operation of at least one vehicle, comprising obtaining data corresponding to at least one vehicle operating characteristic from an on-board unit on the vehicle, providing a plurality of modular applications that are selectable by the user to create a customized system, and processing the data corresponding to at least one vehicle operating characteristic obtained via the on-board unit according to at least one function associated with at least one selected modular application.

[0009] Yet another embodiment is directed to a computer system having an application program, wireless communication framework for processing messages provided by the application program, and a plurality of transport modules

that link the wireless communication framework to a respective plurality of networks for transporting the message to a second unit.

[0010] A method directed for transporting such messages from a first unit is also provided. This method may include the following. The message is first transported from an application program to a wireless communication framework. The message is then processed in the wireless communication framework to select one of a plurality of transport modules that correspond with one of a plurality of networks. The message is then transported through the selected network to a second unit.

[0011] In another embodiment, a method for dispatching a message from a first unit and receiving a message at a second unit is provided. Here, the first unit includes a first application program and a first part of a wireless communication framework. The second unit includes a second application program and a second wireless communication framework. The message is dispatched from the first application program and received in the first part of the wireless communication framework. The message is processed to select one of a plurality of transport modules that correspond to one of a plurality of networks. The

message is transported through the network to the second unit. The message is received in a second part of the wireless communication framework and processed for the second application program. The message is provided to the second application program by the second part of the wireless communication framework.

[0012] Further embodiments and variations of the invention will be apparent from the drawings and description below.

BRIEF DESCRIPTION OF DRAWINGS

[0013] Exemplary embodiments are described below in conjunction with the appended drawing figures, wherein like reference numerals refer to like elements in the various figures, and wherein:

[0014] Figure 1 is a representative functional block diagram illustrating an overall system according to one embodiment of the present invention;

[0015] Figure 2 is a representative block diagram illustrating a system architecture according to one embodiment of the present invention;

[0016] Figures 3A and 3B are representative block diagrams of one embodiment of an on-board unit in one embodiment of the present invention;

[0017] Figure 4 is a representative block diagram of a wireless

communication system according to one embodiment of the present invention;

[0018] Figure 5 is a representative block diagram of a wireless communication framework for a wireless communication system according to one embodiment of the present invention;

[0019] Figure 6 is a flow chart depicting the operation of a wireless communication system according to one embodiment of the present invention;

[0020] Figure 7 is another flow chart depicting the operation of a wireless communication system according to one embodiment of the present invention;

[0021] Figure 8 is yet another flow chart depicting the operation of a wireless communication system according to one embodiment of the present invention;

[0022] Figure 9 is a block diagram of a parameter retrieval process according to one embodiment of the present invention;

[0023] Figure 10 is a block diagram of a parameter retrieval process according to another embodiment of the present invention;

[0024] Figure 11 is a block diagram of a parameter retrieval process according to yet another embodiment of the present

invention;

[0025] Figure 12 is a graph illustrating the operation of a threshold alert process according to one embodiment of the present invention;

[0026] Figure 13 is a block diagram illustrating the operation of a tamper alert process according to one embodiment of the present invention;

[0027] Figure 14 is a block diagram illustrating a parameter change process according to one embodiment of the invention;

[0028] Figure 15 is a block diagram illustrating one possible architecture for a remote diagnostics application to be used in one embodiment of the present invention; and

[0029] Figure 16 is a block diagram illustrating one possible architecture of a leased vehicle management application to use in one embodiment of the present invention.

DETAILED DESCRIPTION

1. SYSTEM FUNCTIONALITIES AND ARCHITECTURE

[0030] Figure 1 is a representative functional diagram of a vehicle monitoring and diagnostics system 100 according to an exemplary embodiment. Generally, the inventive system 100 allows monitoring and control of a vehicle fleet by

displaying and controlling data according to a user's customized specifications. The system 100 is designed with modular applications that interact with core data and services so that vehicle parameters can be monitored, analyzed and displayed in a format that is meaningful to a particular user and/or a particular industry. This flexibility allows different users and/or industries to use the same overall system 100 for vehicle and component monitoring despite their disparate vehicle data requirements.

[0031] Referring to Figure 1, the system 100 may include an application service provider (ASP) infrastructure 102 that acts as a gateway between a plurality of vehicles 104, each vehicle having an associated on-vehicle computer (e.g., an on-board unit or "OBU"105) and customizable interface 106. The interface 106 allows a user or machine 106a to select among various applications, such as third-party applications 108 as well as original, system-supplied applications 110, to obtain and analyze various data from the vehicles 104. The applications may include, for example, tools for obtaining real-time fleet characteristics, trend analysis and diagnostics, to perform manual, dynamic or rule-based configuration, as well as allow fleet managers to provide real-time driver/fleet notification. To

ensure that the user receives data that is meaningful to the user's specific application, the user interface 106 can be customized to operate applications selected by the user. In the example shown in Figure 1, different types of users 106a may select different applications as a customized application group 112 to accommodate their specific data monitoring and reporting needs applicable to their own business.

[0032] For example, as illustrated in Figure 1, a dealer/repair facility may select from the offered applications 108, 110, vehicle configuration, scheduled maintenance, remote diagnostics, and concierge services as its application group 112, while a truck manufacturer may select a different collection of applications 112, such as warranty service/campaign support, vehicle history, and guided diagnostics. By offering a variety of modular applications 108, 110 that can be selected and combined according to the needs of a particular user, the same infrastructure 102 can be customized and used by different users for different purposes with little or no modification of the infrastructure 102 itself. Further, by allowing users to access third-party applications 108 through the same infrastructure as system supplied applications 110, the system 100

can leverage services not provided by the system 100 it-self, further increasing flexibility and adaptability.

[0033] Further, in an embodiment of the inventive system using an ASP-based model, an application service provider pro-vides and allows access, on a subscriber basis, to a re-mote vehicle diagnostics, monitoring, configuration and reprogramming tool via the Internet. That is, the applica-tion service provider provides the hardware (e.g., servers, an on-vehicle computer) and software (e.g., database) in-frastructure, application software, customer support, and billing mechanism to allow its customers (e.g., fleet man-agers, vehicle distributors, vehicle dealers, original equip-ment manufacturers ("OEMs"), leasing/rental companies, and the like) to remotely access the vehicles within a fleet. The tool can be used by subscribers to select and access the modular applications 108, 110.

[0034] Note that an ASP-based model can eliminate the need to physically distribute software to users. Instead, new fea-tures and updates can be immediately available to users because the system resides and runs on an application server. In one embodiment, applications that are not on the application server can reside on the OBU 105. The on-board unit applications can be loaded onto the OBU 105

during vehicle installation, and additional applications or application updates can be downloaded onto the OBU 105 through a wireless network connection.

[0035] Figure 2 is a representative block diagram of system architecture 200 according to an exemplary embodiment. The system architecture shown in Figure 2 is one possible way for carrying out the functionalities described above and shown in Figure 1. In this example, the architecture includes three primary components: the interface 106, a system server 202, and the on-board unit (OBU) 105. All three components 106, 202, 105 are designed to communicate with each other through any known means, such as via wireless networks 206(1-3).

[0036] The interface 106 can be, for example, a user interface and/or a machine interface that allows a human or machine to access the infrastructure 102, which includes the system server 202. The system server 202 may include, for example, a series of servers that perform web page hosting, run applications, perform data storage, and/or perform wireless communications network management. In this example, the system server 202 includes a web/application server 202a, a vehicle server 202b, and a communications server 202c, all of which are linked to a

database server 202d. As shown in Figure 2, the system server 202 acts as a link between a web based client (user) 106 with the OBU 105, allowing user access and control to a vehicle data stream via the Internet 210 or other Internet working system.

[0037] The OBU 105 accesses data from various vehicle components and may also generate vehicle data of its own to provide to the system server 202. The OBU 105 may include a wireless communication module l05a to provide a communication link to a wireless network, a vehicle data processing module 105b to process data obtained from the vehicle components, and a vehicle interface 105c connected to, for example, the vehicle data bus to gather data from the vehicle components for processing and managing time or process-critical functions with the vehicle systems, such as electronic control units. The OBU 105 may also include a global positioning system and a driver display interface.

[0038] Each of the system architecture components will be described in greater detail below.

(A) INTERFACE

[0039] The interface 106 may be a standard browser interface and/or a machine-to-machine interface. In the browser

interface, a human user accesses the system via a standard web browser. In one embodiment, the user gains access to the specific set of their authorized vehicles and functions after login and password authorization. In a machine-to-machine interface, server and vehicle data and functions may be accessible via a secure application program interface (API). A machine-to-machine interface allows other applications access to the system's 100 capabilities so they can gain remote access to the vehicle and the capabilities offered by the system. This allows the system 100 to interface with existing or planned back office applications and operations, making the system 100 suitable for integration with, for example, overall fleet operations or other systems.

(B) SERVER

[0040] The server system 202 is the fixed-based component of the system 100, and as explained above, can be an Internet-based system and use an ASP model. The end user can access the system from the interface 106, such as any commercially available web browser. As noted above, the server system 202 in this embodiment includes the web application server 202a, the vehicle server 202b, and the communications server 202c, and the database server

202d. Each of these will be described in greater detail below.

(I) APPLICATION SERVER

[0041] The web application server 202a contains the logic defining one or more applications to an end user. All the data needed for a specific user application is requested and sent to the OBU 105 via one of the wireless communication networks 206(1-3). As will be explained in greater detail below, applications perform the necessary calculations and then package the results for presentation in a defined format to the user. Further, the web application server 202 is responsible for running business applications related to user activities, which may include performing business logic, interfacing to the systems databases for fleet, vehicle, component, and transaction activity, knowledge-base storage, and sending user requested vehicle queries or functions to the vehicle server 202b and the communications server 202c.

(II) VEHICLE SERVER

[0042] The vehicle server 202b stores and processes vehicle-specific data and acts as a translator between the applications 108, 110 and the specific vehicle and/or vehicle

component. More particularly, the vehicle server 202b is responsible for processing data requests and vehicle responses and converting the outbound and inbound data into translated vehicle information.

[0043] The vehicle server 202b translates user requests from the interface 106 into formats specific to the vehicle 104 to which the request is directed. The vehicle server 202b conducts this translation without any user interaction or property selections. For example, the vehicle server 202b may evaluate a message being sent to a particular vehicle and detect the vehicle type, the vehicle bus type, and the vehicle component or sub-component that is intended as the message recipient. The vehicle server 202b then packages the message according to the specific communication protocol mandated by the recipient component. As a result, the vehicle server 202b allows monitoring and control of different vehicles having different components through the same interface 106 for a given user and application.

(III) COMMUNICATION SERVER

[0044] As shown in Figure 2, one embodiment of the inventive system allows communication between at least the vehicles 104 and the server 202 via a wireless network, such

as a satellite or terrestrial based network. A communication server 202c may be included in the server 202 to support wireless communications and provide a central location for supporting changes and improvements in wireless technology. In one embodiment, the communication server 202c manages the communications activities between the OBU 105 and the vehicle server 202b and processes vehicle/component specific requests between the OBU 105 and the vehicle server 202b.

[0045] In one embodiment, the communications server 202c utilizes a wireless communications framework (WCF) that provides a communication link between the system server 202 and the vehicle 104. Although any wireless mobile communication system can be used in the inventive system 100, a flexible wireless communication infrastructure that is capable of handling multiple platforms and/or multiple communication providers is preferred. One possible embodiment of such a framework is described in U.S. Provisional Application No. 60/35 1,165 (Attorney Docket No. 65855-0060), entitled "Wireless Communication Framework", filed January 23, 2002, the disclosure of which is incorporated herein by reference in its entirety and described in more detail below.

[0046] To handle multiple communication providers and/or platforms, the flexible wireless communication infrastructure may abstract the needs of a specific wireless communication provider, such as the message size, message format, and specific protocol details, into a standard messaging API understandable by multiple systems and platforms. In one embodiment, the communication server 202c abstracts messages, and stores and forwards messages to ensure later delivery of messages to vehicles that are temporarily outside a wireless communication coverage area, and may even include least cost routing rules to select among multiple wireless networks to prioritize message routing based on cost and/or criticality of the message.

(IV) DATABASE SERVER

[0047] The system server 202 also includes a database server 202d containing relational data tables designed to retain information pertaining to a user, a vehicle, a fleet, system transaction history and other relationships associated with a given vehicle 104. The database server 202d also may be designed to retain the data resulting from any vehicular transaction, such as transactions between the OBU 105 and the system server 202. In one embodiment, the database is structured such that authorized users can ac-

cess vehicles in a number of ways, for example, by fleet ownership, leasing fleet, vehicle manufacturer, and component manufacturer. This structure enables the system 100 to provide each of these beneficiaries with specific, customized data and access in a format meaningful to each user.

(c) COMMUNICATION NETWORKS

[0048] As noted above, the server system 202 and OBU 105 can communicate via one or more communication networks. Each of the communication networks may be a partial or full deployment of most any communication or computer network, and thus, can include a few or many network elements, most of which are not shown. Each communication network may include circuit-switched as well as packet-data elements to provide transport of at least data communications between server system 202 and OBU 105. It can be public or private, terrestrial wireless or satellite, and/or wireline, such as the wireless communication networks 206 (exemplified by wireless communication networks 206(1-3).

[0049] Public wired and/or wireless networks typically provide telecommunications and other networking services in a particular geographic coverage area to its subscribers.

And generally, any interested member of the public meeting minimal criteria may become a subscriber of public network. In the case of wireless or satellite networks, public networks typically provide coverage to other wireless networks" subscribers who are roaming within the coverage area of network as well.

[0050] Additionally, the coverage area of public network is typically wide-ranging. For example, the coverage area of a wireless public network may encompass a metropolitan area, a substantial part of a metropolitan area, numerous metropolitan areas, or an entire nation or region. When integrated with public wired and satellite networks, the combined networks provide national along with international coverage. Thus, each of the wireless communication systems 206 may include portions of a Public Switch Telephone Network (PSTN), the Internet, core and proprietary public networks, and/or wireless voice and packet-data networks (e.g., 1G, 2G, 2.5G and 3G telecommunication networks).

[0051] Each communication network may be a private or "enterprise" wired or wireless network as well. Unlike public networks, private networks advantageously provide the enterprise with greater control over the network, which in

turn allows vast customization of the services (e.g., localized abbreviated dialing) provided to the network's users and/or subscribers.

[0052] These networks are "private" in that the networks" coverage areas are more geographically limited. Typically, but not necessarily, subscription to such private networks is limited to a select group of subscribers. Networks deployed by many enterprises that only allow their employees, customers, vendors, and suppliers to subscribe exemplify these private networks.

[0053] Many enterprises, Small Office/Home Office (SOHO) entities, and private individuals have private-wireline-switching systems. These private-wireline-switching systems may include, for example, private branch exchanges (PBXs) and/or media gateways. The private-wireline-switching systems switch, couple or otherwise connect communications (i) internally, i.e., among the subscribers of the network and (ii) externally, i.e., between the subscribers of the private network and subscribers of public networks.

[0054] In addition to the wireline networks, enterprises, SOHOs and private individuals are increasingly deploying private wireless communication systems, such as wireless office

telephone systems ("WOTS") and/or wireless local area networks (WLANs), e.g., Bluetooth and/or IEEE 802.11 WLANs, in lieu of or in addition to wireline switching systems. Similar to public networks, private networks may be integral to or integrated with other private and public satellite, terrestrial wireless, and wireline networks to provide national and international coverage.

[0055] Accordingly, each of the wireless communication networks 206 can be any of a private and/or public terrestrial, satellite and/or other wireless network. And each of the wireless communication networks 206 can be incorporated into a wide-area network (WAN), thereby creating a Wireless WAN (WWAN); a local area network (LAN), thereby creating Wireless LAN (WLAN); and/or other wired network. Alternatively, each of wireless communication networks can be a standalone WLAN.

[0056] At times, a single wireless service or technology may not provide a desirable messaging cost structure or geographical coverage to support all the features and users of the system 100 (Figure 1). Multiple services might be used to provide for dynamic balancing between messaging costs and message timeliness. In addition, different wireless services and technologies may have different applica-

tion program interfaces (APIs) and/or require custom interfaces for given computing environments. Moreover, messaging capabilities may differ across different services and technologies.

(D) ON BOARD UNIT (OBU)

[0057]   As noted above, the OBU 105 provides the vehicle-side, real-time computing base for the system. In one embodiment, the OBU 105 is responsible for data stream processing, discrete measurements, vehicle position information, wireless communications, and real-time analysis of data. Within the system's overall framework, the OBU 105 acts as a vehicle server, providing vehicle specific data and functionality. In one embodiment, the OBU 105 may be an expandable custom hardware platform designed and manufactured to reside on a wide variety of vehicles with different component specifications and needs and is preferably capable of running multiple applications while acting as a vehicle data gateway for others.

[0058]   Figures 3A and 3B are representative high-level block diagrams of the OBU 105. One embodiment of the OBU 105 may include a data processor 300 and one or more interfaces 302, 304, and 306. Included among the interfaces is a wireless interface 302 that may control communication

between the OBU 105 and the system server 202 via one of the wireless networks 206(1-3), a vehicle interface 304 that allows the OBU 105 to transmit to and receive from, for example, electronic control units (ECUs), vehicle controllers, and/or other vehicle components 308, and an optional user interface 306 that allows a user to read and/or enter information into the OBU 105.

[0059] The wireless interface 302 in one embodiment sends and receives data from the system server 202 to and from the OBU 105 and abstracts communication operating parameters from different wireless network devices to allow the OBU 105 to communicate with a flexible wireless communication infrastructure, such as the type described above with respect to the system server 202. More particularly, the wireless interface 302 may encapsulate protocol differences among various wireless network devices to provide a standard output to the processor 300. In one embodiment, wireless network messages are routed from the system server 202 to the wireless interface 302 for error checking and filtering. After filtering, commands are passed to the processor 300 and then routed to their respective vehicle components.

[0060] The processor 300 acts as the central processing unit

(CPU) of the OBU 105 by managing the sending and receiving of requests between the system server 202 and the vehicle 104 via the wireless interface 302. In one embodiment, the processor 300 has the logic and intelligence to carry out vehicle specific services such as diagnostic requests and processing. For example, the processor 300 may run specific applications that are loaded into the OBU's memory 315, 316, 318 (Figure 3B) and coordinates activities between the vehicle datastream, GPS unit 313 (Figure 3B), wireless communications 302, and the system server 202. Further, in one embodiment, the processor 300 can be updated through the one of the wireless networks 206(1-3) to add and enhance its functionality. This capability eliminates requiring the vehicle to be at the service bay for software updates that enhance features and functionality.

[0061] The vehicle interface 304 allows the OBU 105 to support a wide variety of vehicle components and subcomponents. Possible interfaces that can be supported by the OBU include SAE JI708, SAE J1939, SAE OBDII/CAN, ISO-9141, discrete I/O, proprietary interfaces, and other interfaces (e.g., discrete or instrumentation interfaces). Further, the vehicle interface 304 provides a single point of contact for

all vehicle components and control devices on the vehicle 104, allowing the communication between OBU software with the vehicle's actual data bus line as well as wireless communication devices, such as a satellite-based communications system.

[0062] In one embodiment, the vehicle interface 304 may include a data parser/requester module 310 that contains software code logic that is also responsible for handling direct interfacing between the processor 300 and the vehicle data bus 307 for non-application specific (i.e., "generic" SAE J1708 or SAE1939 discrete measurement points) parameter readings, alerts, configuration or reprogramming data, as explained in greater detail below.

[0063] The vehicle interface 304 may also include, for example, one or more application specific modules 312, such as one for each manufacturer specific controller 308 within the vehicle 104, each containing software code logic that is responsible for handling interfacing between the processor 300 and the vehicle data bus 307 (via data parser/requestor module 310 in this example) for application/specific parameter readings, alerts, configuration or reprogramming data.

[0064] Note that the OBU 105 may act as a server and/or data

gateway for an application that places data directly on the vehicle data bus 307. In one embodiment, the OBU 105 uses an interface standard, such as TMC RP 1210A, as an element of the data gateway. Regardless of the specific standard used, any activity using the OBU 105 as a data gateway will involve data going through the processor 300.

[0065] In an embodiment of the present invention, the OBU 105 is designed to be compliant with the SAE's *Joint SAE/TMC Recommended Environmental Practices for Electronic Equipment Design (Heavy-Duty Trucks)*, Document No. Jl455 (August 1994) standard, which is incorporated herein by reference in its entirety, because it will be a component included (or installed) within vehicles 104. As indicated above, the OBU 105 is not limited to be compliant with any particular standard and can accommodate any on-board electronic system standard (e.g., SAE Jl708, SAE Jl939, SAE J1850, ISO 9141, proprietary data streams, etc.) for any sub-system (e.g., engines, transmissions, braking systems, in-strument clusters, etc.) as long as the system 100 is capa-ble of converting commands between the interface 106 and the OBU 105 according to whatever standard is used by a given vehicle electronic system. If the vehicle elec-

tronic system uses a proprietary standard, for example, the vehicle server 202b and the associated application specific module 312 on the OBU 105 may be adapted to accommodate the proprietary standard.

[0066] Figure 3B illustrates another embodiment of the OBU 105 and explicitly shows the capability to interface with other devices via, for example, a parallel interface, serial interface, universal serial bus (USB), satellite, terrestrial wireless (e.g., 802.11b), and/or a global positioning system (GPS). More particularly, the embodiment of the OBU 105 shown in Figure 3B includes a GPS circuit 313 that receives signals from a GPS antenna and a serial interface 314 that communicates with a driver interface or other on-vehicle devices (not shown), such as a handheld device, a cellular telephone, voice messaging system, data logger, or other devices. Figure 3B also explicitly illustrates a flash memory 315, a dynamic random access memory 316, and an optional compact flash memory 318 coupled to the processor 300 as well as a power supply 320 coupled to the vehicle battery and ignition switch (not shown). Those of skill in the art will understand that the elements and concepts shown in Figures 3A and 3B can be combined in any manner without departing from the

scope of the present embodiment.

[0067] The application software and the application framework are built with both a software and hardware abstraction layer. This approach makes the framework adaptable to a number of alternative operating system and hardware platforms. One embodiment of the OBU 105 may use any known real-time operating system.

(E) WIRELESS COMMUNICATION FRAMEWORK

[0068] Figure 4 illustrates exemplary architecture 400 of the wireless communication framework (WCF) in accordance with an exemplary embodiment. The WCF may encompass a number of software and/or hardware program and/or application elements for carrying our wireless communications between two wireless nodes.

[0069] The WCF architecture may be concentrated on either the server system 202 or a remote unit that is operable to communicate with the server system 202, such as the OBU 105. In such case, the device having server portions of the WCF architecture may act as a server in a client/server relationship and the device having the client portions may act as the client. Because the server and client responsibilities may change depending on the function to be carried out, the server system 202 can be a server for one

function, yet a client for another. Similarly, the remote unit may be a client for one function, and a server for another.

[0070] Alternatively, the WCF may be distributed among one or more server-system elements 402 and one or more remote-unit elements 404. In a distributed embodiment, the remote-unit elements 404 may be included within a remote unit, such as the OBU 105, and the server-system elements 402 may be deployed in the server system 202.

[0071] In addition to the OBU 105, it is understood that the remote unit may a Personal Digital Assistant (PDA) and/or another computer that can be communicatively coupled with server-side elements 402. As such, the remote unit may contain server-system elements 402 in addition to or in lieu of the remote-unit elements 404, thereby enabling communications between itself, the server-system 202 and/or another remote unit, such as another OBU (not shown).

[0072] The remote-unit elements 404 may include (i) remote-unit application programs 406a (e.g., full or partial application programs that reside on the remote unit) such as those as described above, (ii) remote-unit transport modules 410a, and (iii) one or more intermediary remote-unit

WCF modules 408a.

[0073] Similarly, the server-system elements 402 may include (i) server-system application programs 406b, which may or may not be counterparts to the remote-unit application programs 406a; (ii) server-system transport modules 410b, and (iii) one or more server-system WCF modules 408b, which may or may not be the same as the remote-unit WCF modules 408a.

[0074] With reference to Figure 4, the transport modules 410a, 410b, alone or as a complementary pair, may be deployed as one or more different software programs, software modules, and/or hardware modules for connecting and interacting with various communication networks, such as the wireless communication networks 206(1-3). The transport modules 410a, 410b provide one or more interfaces through which the application programs 406a, 406b couple to the WCF modules 408a, 408b.

[0075] In doing so, each of the transport modules 410a, 410b may be configured to (i) execute protocols to access its corresponding communication network, (ii) initialize, maintain, and shut down server-system and/or remote unit communication adapters (e.g., modems), respectively, (iii) test the server-system and/or remote unit communi-

cation adapters, and/or (iv) perform any other functions and/or procedures to carry out communications with the corresponding communication network for which the particular transport module 410a or 410b is associated.

[0076] Each of transport modules 410a, 410b may be tailored (e.g., abstracted or otherwise configured) for access to a specific implementation and/or format of a communication network. If, for example, each of the wireless communication networks 206(1-3) are different from each other, then a first of the transport modules 410a, 410b may be configured to access wireless communication network 206(1), a second may be configured to access wireless communication network 206(2), a third may be configured to access wireless communication network 206(3), and so on. Other transport modules 410a, 410b can be included to access additional network services, such as those provided by WLANs or WWANs.

[0077] The number of transport modules 410a, 410b deployed in the remote-unit and/or server-system elements 402, 404 may be a function of the number, configuration and/or format of the communication networks 206(1-3) that the server-system 202 and/or the remote unit may have access to. For instance, the remote unit might not need to

have access to the Internet. And thus, having a transport module 410a for Internet access may be omitted. On the other hand, the server-system elements 402 may have access to a host of networks that the remote unit elements 404 do not. Accordingly, the server-system elements 402 may have transport modules 410b configured to carry out communication with these networks.

[0078] If, for example, remote-unit elements 402 are deployed in a number of remote units in a fleet of vehicles that travel in a specific geographical region where access is available to wireless communication networks 206(1), 206(2), but not wireless communication network 206(3), then the remote-unit transport modules 410a may be configured to access the wireless communication networks 206(1), 206(2).

[0079] The server-system elements 404, which may or may not be co-located in the same specific geographical region as the remote units, may have access to the wireless communication network 206(3) (e.g., a WLAN) in addition to the other communication networks. Thus, the server-system transport modules 410b may be configured to access wireless communication network 206(3) as well.

[0080] Thus, the server-system elements 404 may have access to

a host of networks to communicate with each of the remote unit elements 402, even though not all of the remote unit elements 402 have the corresponding access. For instance, one of the remote unit elements 402 may have access to only network 206(1), while another of the remote unit elements 402 may have access to only network 206(2), but the server system elements 404 may have the transport modules 410a, 410b to support both networks 206(1-2).

[0081] Moreover, the number of transport modules 410a, 410b may be a function of the monetary and overhead costs of subscribing to multiple communication networks. For instance, the number of transport modules 410a, 410b may be limited when monetary cost savings (e.g., discounted delivery rates) in using more networks cannot be justified. Other non-monetary costs, such as memory usage and processing losses, may also limit the number of transport modules 410a, 410b.

[0082] Conversely, the number of transport modules 410a, 410b may be greater when non-monetary and monetary cost savings can be obtained by the use multiple networks. These cost savings may be embodied as discounted rates, which can be apportioned by time, volume of calls; time of

day, month, etc; geographical region; and other metrics.

2. WIRELESS COMMUNICATION MODULES

[0083] Figure 5 illustrates the WCF modules 408a, 408b in greater detail. The WCF modules 408a, 408b may be deployed with a messaging Application Program Interface (messaging API) 512, a message manager 514, a transport manager 516, message-storage manager 518, a message store 520, ordered delivery manager 522, a least-cost routing manager 524, and a multi-part message manager 526.

[0084] Regardless of whether the WCF is distributed among or concentrated on a remote unit and/or server system 202, some or all of the functions of WCF modules 408a, 408b may be deployed in both the remote-unit and server-system elements 402, 404. In the case of a concentrated system, however, function calls may be used to establish communication between the concentrated elements.

[0085] In some instances, the remote-unit WCF module 408a might not perform the same functions that are carried out by the server-system module 408b, but rather it may perform functions that complement and/or accompany functions carried out by the server-system elements 408b. Similarly, the server-system WCF module 408b might not

perform the functions that are carried out by the remote-unit module 408a, but rather functions that complement and/or accompany functions carried out by the remote-unit elements 408a. Further, some of the functions of the WCF modules 408a, 408b may be applicable to only a remote unit or server system 202. Thus, some of the functions carried out by WCF modules 408a, 408b may only exist in either the remote unit or server-system elements 402, 404. Further detail of the wireless communication framework modules 408a, 408b is provided below.

(A) MESSAGING API

[0086] The messaging API 512 may provide the functionality to receive, send, and process messages sent to or coming from multiple application programs. This functionality can be carried out by the messaging API 512 even if the application programs 406a, 406b use program and data structures different from rest of the WCF architecture.

[0087] As such, the messaging API 512 may allow many different application programs to use a common and/or "standardized" messaging format provided by the WCF modules 408a, 408b. This beneficially allows the development of application programs without the need for custom or tailored programming. For instance, the WCF architecture

and the messaging API 512 may provide the messaging system, bridge (gateway, and/or conduit), storage, and programming that allow an application program to be developed and implemented independent of the communication network used by the WCF architecture.

[0088] In facilitating such application development independence, the messaging API 512 may abstract the differences between transport providers (e.g., the providers of wireless communication networks 206(1-3)) and the differing technologies of the wireless communication networks to allow applications to be written independently of the services and transport providers selected. Accordingly, when a new application program is to be added, the messaging API 512 may provide hooks or other access interfaces to the application programs to achieve communication without substantial custom programming.

(B) MESSAGE MANAGER

[0089] The message manger 514 may manage the delivery and acceptance of messages to and from the application programs 406a, 406b. The message manager 514 may also manage a reliable delivery function that insures messages are not dropped or lost. The reliable-message delivery performed by the message manager 514 may be accom-

plished using message-delivery verification, which will be discussed in greater detail below.

(c) TRANSPORT MANAGER

[0090]   The transport manager 516 manages the selection of transport modules 410a, 410b available to the remote unit and/or server system elements 402, 404. The transport manager 516 may work in conjunction with other components of the WCF modules 408a, 408b, such as the least-cost routing manager 524 (discussed below) for determining which of the transport modules 410a, 410b to select.

(d) ORDERED DELIVERY MANAGER

[0091]   The ordered delivery manager 518 manages an ordered delivery of messages sent by any of the application programs 406a, 406b. Ordered delivery may be any predefined order in which messages are to be sent, and can be configured as an ordered delivery service. This provides a significant advantage when performing database edits or other information that is order dependent.

[0092]   When ordered delivery is desired or requested by any of the application programs 406a, 406b, the order delivery manager 518 may arrange the messages in any of the

plurality of predefined orderings irrespective of when the WCF modules 408a, 408b receive the messages from the application programs 406a, 406b. Under the WCF messaging system, messages can be configured to specify a given delivery queue using, for example, a queue identifier or other notation. Under this scheme, messages with a given queue identifier may be sent to a particular queue, as indicated by the queue identifier. Before transmission, the messages may be arranged in the particular ordering selected.

(E) LEAST-COST-ROUTING MANAGER

[0093] The least-cost-routing manager 524 may be responsible for selecting one or more of the transport modules 410a, 410b based on a plurality of factors, such as the cost and timeliness of message delivery. This WCF module may be expanded using custom routing factors as desired.

[0094] In addition to cooperating with other WCF modules 408a, 408b, the least-cost-routing manager 524 may operate in combination with the transport manager 516 to determine which of the transport modules 410a, 410b to select. The least-cost routing manger 524 may, for example, link or relay information to the transport manager 516 after determining the low cost provider so that the messages may

be transmitted via the low cost service provider.

(F) MULTI-PART MESSAGE MANAGER

[0095] The multi-part message manager 526 may manage disassembly (and reassembly of previous disassembly) of large messages for transport among the server system 202 and any of the remote units. This is particularly advantageous when the message to be sent exceeds the maximum allowable message size for the selected one of the networks 203(1-3). The multi-part message manager 526 may be invoked to ensure that the message adheres to the set maximum message size by disassembling the message into groups of smaller chunks.

[0096] The chunk size may be, for example, 16 or 32 byte chunks. Chunk size selection may also depend upon the maximum allowable size message that can be sent over the selected one of the wireless communication networks 206(1-3) without degradation or loss of the contents of the message. The chunk size may be based upon satisfactory transmission accuracy returned from error-control algorithms, for instance. The multi-part manager 526 may be equipped with intelligence, e.g., dynamic and/or statistical algorithms, for selecting a proper chunk size for maximizing throughput, bandwidth and/or other trans-

mission parameter.

[0097] The following describes, by way of a simple, non-limiting example, of how the multi-part manager handles such overage. Assume for this example that the wireless communication networks 206(1) may have a maximum allowable message size of about 100 bytes per message, and the network 206(2) may have an allowable message size of about 512 bytes per message. In either case, however, a certain number of bytes per chunk, e.g., 2 bytes, may be allocated to overhead, reducing the number of available bytes for transmission of content over wireless communication networks 206(1-3).

[0098] Assume for this example that a message having content equaling about 100 bytes is destined for transmission across the wireless communication networks 206(1). Since the content of the message exceeds the 100 byte maximum allowable message size, the multi-part message manager 526 can disassemble the message into the smaller 16 and/or 32 byte chunks.

[0099] If, for example, the 16 byte chunk is selected, the chunk size dictates that the message will be first broken down into five 16-byte chunks, totaling 80 bytes (16 X 5 = 80) + 10 bytes overhead. At this chunk size, another 16 byte

chunk cannot be sent because the sixth 16-byte chunk causes the accumulated byte size to equal 106 bytes (plus 2 bytes overhead), thereby exceeding the maximum allowable message size of 100 bytes by 8 bytes. The multi-part message manager 526 can reassemble the five chunks into a first-part message, leaving the 10 remaining bytes (100 90) for a second-part message, which may be have other content added to optimize available bandwidth. In such case, the first-part message will only have 10 unused bytes.

[0100] If, on the other hand, the larger 32 byte chunk size was selected, then the chuck size dictates that the first-part message will be broken down into only two chunks (2 X 32 = 64) + 4 bytes overhead. A third chunk will cause the accumulated byte size to exceed the maximum allowable message size of 100 bytes by 2 bytes ((3 X 32) + (3 X 2) = 102). Consequently, the first-part message would have 32 bytes of unused space, which is 22 bytes more than when using the 16 byte chunk size. Accordingly, the smaller chunk size allows more of the available message size to be utilized.

[0101] The added number of smaller chunks, however, may cause more chunks to be sent. This may increase the amount of

overhead, which may accumulate as a function of the number of chunks. With smaller message sizes, not many chunks need to be sent. In some circumstances, an optimization penalty resulting from overhead incurred when sending a smaller message with smaller chunks might not outweigh the reduction of unused bytes accomplished with the smaller chunk size.

[0102] When using the second wireless communication networks 206(2), for example, the multi-part message manager 526 may more optimally disassemble the message using the larger 32 byte chunks. As noted, the increased number of chunks required to be sent to accommodate the 512 byte size message increases the amount of corresponding overhead needed to send the message. Such an increase may fail to outweigh the reduction of unused bytes that accompanies the use of smaller chunks. As is apparent, larger chunk size may be beneficial for larger message sizes, the smaller chunk size may be more beneficial for smaller available message sizes. As such, a programmer or user of the WCF can flexibly pick a predetermined maximum byte size limit of a network at which the multi-part message manager 526 will disassemble a message into smaller or larger chunk sizes. The user, for example,

could select a prescribed threshold of 200 bytes, which may allow a message to be disassembled into smaller chunks only when the maximum allowable limits falls below this threshold. Messages otherwise may be disassembled into larger chunks when the maximum allowable limits rises above this level.

[0103] The multi-part message manager 526 can also break messages into chunks for other reasons than to accommodate large messages. For instance, the chunk size may be set to a size slightly smaller than the maximum allowable byte size of the wireless communication networks 206(1-3) regardless of actual message size. With this alternative and with smaller chunk sizes in general, the probability that a message will get through a network without unacceptable retry attempts increases.

[0104] Additionally, the multi-part message manager 526 might not need to re-send already-acknowledged message chunks even if a partially complete message is re-routed through another network (e.g., from network 206(1) to network 206(2)). Accordingly the chunk size may be changed to another size based on which networks are available. If, for example, the message is rerouted from a network 206(2) to network 206(1), the chunk size may

change from 32 bytes to 16 bytes. However, the chunk size in one network may be compatible with another, and thus, need not to be changed. Other variations on the size of the chunk in relation to the network may be utilized as well.

(G) MESSAGE STORAGE MANAGER

[0105] The message-storage manager 518 may be responsible for keeping a queue of messages that are waiting to be sent, have been sent or are awaiting an acknowledgement. In the former case, the message-storage manager 518 may operate in conjunction with other WCF modules 408a, 408b to maintain the message in the queue until one of the transport module 410a, 410b connects to the chosen network.

[0106] In the latter case, the message-storage manager 518 may maintain a record (e.g., a table) of sent messages. This record may be used to ensure reliable delivery of the message in case of a failure of system 100, an out-of-coverage area error, and/or other error. With the record, a message is able to be resent from the message store 520 in response to such a failure.

[0107] At the receiving end, the message-storage manager 518 may also maintain a copy of the message as a handshak-

ing mechanism. This copy may be maintained until the message is successfully delivered to the target application. If, for example, the server system 202 sends to OBU 105 a message when the vehicle 104 is outside the coverage area of any of the networks 206(1-3), then the message-storage manager 518 may store the message in the message store 520. After the vehicle 104 enters the coverage area of one or more of the networks 206(1-3) .the message may then be sent.

[0108] If the multi-part message manager 526 has disassembled the message into chunks, then the chunks already sent may be stored at the message store 520 of the OBU 105, and the chunks not sent may be maintained at the server system 202. This beneficially prevents unnecessary and costly retransmission of already sent chunks in the case of, for example, system failure, out-of-coverage area errors or other connectivity failure. Such benefit may be realized because only the chunks maintained in the message store 520 of the server system 202 are sent to the OBU 105. Since the chunks that have been already sent may be maintained in the message store 520 of the OBU 105, the message can be reassembled using the earlier stored chunks and the chunks sent after reconnection.

(H) MESSAGE SERVICES, WCF EXTENSIBILITY, AND WCF PORTABILITY

[0109] The WCF may also carry out certain message services such as encryption, compression and packaging. The WCF may use standard as well as custom encryption algorithms and cryptography over the entire communication route. Different types of encryption services may be used over different segments of the communication route. The encryption services may be used in addition to or in lieu of any encryption provided by the network service providers. Conversely, the WCF may use the encryption services provided by the network service providers in lieu of the services provided by the WCF.

[0110] To limit bandwidth that is required for transmission, messages can be compressed using standard and custom compression algorithms. Different types of compression services may be used over different segments of the communication route. The compression services may be used in addition to or in lieu of any compression provided by the network service providers. Alternatively, the WCF may use the compression services provided by the network service providers in lieu of the services provided by the WCF.

[0111] As noted, the WCF may carry out packaging, which allows

one or more messages to be packed together to reduce the cost of transmission over transports that support large message sizes. Thus, instead of incurring the overhead and/or acknowledgement costs for each message, these costs may be incurred only once for several messages.

[0112] The WCF may be extended in various ways so as to provide extension interfaces that allow the system and/or application designer to customize and extend the WCF for the particular computing platform capabilities and behavior. Included in the extension capabilities are message store, transport modules, least-cost-routing manager, compression, and encryption extensions.

[0113] The message store 520 provides the storage for messaging functions, including message persistence in which messages are maintained message information over a system or component reset, reliable message delivery, order delivery processing, and multi-part messaging. The message store 520 may be customized according to platform capabilities and system requirements.

[0114] In addition to customizing the message store 520, new transport modules 410a, 410b may be added to support additional communication services and providers. The

least-cost routing manager 524 may be customized to provide sophisticated routing and escalation logic.

[0115] As new standards and protocols are developed for message compression and encryption, the WCF may be extended to incorporate the changes. New compression and encryption modules may be used to support non-standard and proprietary protocols.

[0116] In addition to being extensible, the WCF may be ported to between platforms and systems. In one exemplary embodiment, an operating-system-thin layer isolates the WCF from operating system differences, thereby allowing portability between platforms. The message store 520 may abstract the file system, memory structures, and/or database structures in which messages are stored. The WCF may be deployed as dynamic-link libraries or shared libraries on platforms that support such libraries. Alternatively, the WCF may be deployed as static-linked libraries on platforms that support such libraries. The WCF may make use of Computer Object Model (COM) and can be integrated with COM+ on a Windows 2000 platform. As such, the WCF may use the transactional and security features of COM+.

(I) RELIABLE MESSAGE DELIVERY

[0117] As noted, the WCF modules 408a, 408b may bridge or provide a gateway between the application programs 406a, 406b and the transport modules 410a, 410b. In an exemplary embodiment, the WCF modules 408a, 408b can bridge or otherwise couple the remote-unit application programs 404a with the server-system application programs 404b, the transport modules 410a, 410b using standardized and/or proprietary messaging system.

[0118] As noted, the API 512 may manage the acceptance of messages from remote-unit application program 406a and the delivery of messages to server-system application programs 406b. The message manager 514 may also manage a process for carrying out a function for ensuring that messages are reliably delivered (i.e., reliable-message delivery). The reliable-message delivery may include the process for verifying that a sent message was properly received (hereinafter "message-delivery verification"). The response time and deployment of message-delivery verification can be based on the urgency of the message, for example.

(J) EXEMPLARY MESSAGE DISPATCH

[0119] Figure 6 illustrates a flow chart depicting a communication flow 600 between the service server 202 and the OBU

105. The following describes the flow 600 of one or more messages originating from the system server 202 and terminating at the OBU 105. The flow 600, however, may also be carried out in the reverse direction, i.e., originating from the OBU 105 and terminating at the system server 202. Moreover, other devices besides the server system 202 and the OBU 105 may carry out the following flow 600.

[0120] In block 604, one or more of the messages is dispatched to the WCF from one of the application programs 406a. This dispatch may be carried out via messaging API 512. As noted, the messaging API 512 can receive and process messages coming from one or more application programs 406a, 406b. Since the messaging API 512 can select and provide a corresponding interface for the one or more of the transport modules 410a, the applications 406a can be written to communicate with the messaging API 512, thereby allowing a programmer to focus on the vehicle application at hand and not the code for carrying out communications over the wireless communication networks 206(1-3).

[0121] In block 606, the messages are configured and formatted for dispatch. The desired transport module 410b that cor-

responds to the selected one of the wireless communication networks 206(1-3) may be selected. The selection may be based on many factors, such as those described hereinafter. The process of selecting one or more of the transport modules 410b (and in the reverse direction, transport modules 410a) may include reviewing and/or determining network services that are available to the OBU 105. The server system 202 may contain a library of the network services available to one or more of the remote units, such as OBU 105, to assist in the selection of the desired transport module 410b. After determining the available network services, the WCF architecture can proceed to select one or more of the available transport modules 410b for each available network 206(1-3). Other factors, such as cost or transmission speed, may be likewise included in making the determination.

[0122] In block 608, the messages are dispatched through the selected transport module 410b for the desired wireless communication networks 206(1-3). In block 610, the messages are received by the OBU 105 via one of the transport modules 410a that corresponds with the wireless service selected by the server system 202.

[0123] In block 612, the messages are processed by the WCF ar-

chitecture of the OBU 105. This may include the multi-part message manager 526 reassembling messages that may have been disassembled by the server-system elements 404 of multi-part message manager 526. Also, the message storage manager 518 or message manager 514 may store the messages in the message store 520 for later processing. The WCF architecture may also perform other desired processing, as noted above, including formatting the messages for delivery to and receipt by one or more of the application programs 406b. Since many application programs 406b may be run simultaneously or concurrently in the OBU 105, the WCF architecture and functionality has the ability to format the messages to suit a number of different possible application programs 406b. Such formatting may be accomplished through the messaging API 512.

[0124] In block 614, the messages are sent to one or more of the application programs 406b via the messaging API 512. As noted in block 606 described above, the message manager 514 may be used to provide reliable-message delivery. To facilitate the reliable-message delivery, the messages may be assigned a delivery option by one or more of the application programs 406b. This delivery option

may be either unreliable or reliable status. If unreliable status is applied to one or more of the messages, then the message manager 513 may cause the message to be delivered without requiring the OBU 105 to return a confirmation acknowledgement or receipt. In the simplest case, the delivered messages are sent and forgotten. If, however, one or more of the application programs 406b assigns a reliable status to one or more of the messages, then these messages may be repeatedly sent to the OBU 105 until the application programs 406b and/or server system 202 receive a return confirmation acknowledgement or receipt.

[0125] Also noted in above, ordered delivery manager 518 can provide order delivery processing of the messages or message chunks. To facilitate the order delivery, one or more of the application programs 406b may assign a particular order to a plurality of the messages. In this case, the particular assigned order is the order in which the messages are to be sent. The OBU 105 may use the order delivery processing to properly process transmitted information. The ordered delivery manager 518 may collect the messages until the ordered-delivery processing is complete, and then forward the messages to the application

programs 406b. Then, the ordered delivery manager 518 dispatches the messages according to the assigned order. The WCF architecture and functionality supports multiple, independent, ordered delivery queues.

[0126]   Referring now to Figure 7, a communication flow 700 illustrating a dispatch of one or more messages via the messaging API 512 is described in greater detail. The messaging API 512 communicates with the transport module 410b of a desired network 206(1-3) via a transport-send agent of the transport manager 516 to query whether the transport module 410b is allowed to send messages as shown in block 702. This ensures that the OBU 105 is within range to receive messages before any message is dispatched. Determining whether the OBU 105 is within range of the network 206(1-3) may be facilitated using capabilities of the communication hardware and software of the OBU 105, as is known to one skilled in the art.

[0127]   If, for example, the OBU 105 is within range of network 206(1) and/or network 206(2), then the messages may be sent in block 704. If the vehicle is not within range, then block 702 is repeated until the vehicle becomes within range of the corresponding network. During this wait

time, messages may be stored within the message store 520 by message-storage manager 518, thereby freeing up the application program 410a to perform other tasks.

[0128] With continued reference to Figure 7, the operation of the multi-part message manager 526 is described in greater detail. As noted, the multi-part message manager 526 may disassemble large messages that exceed the maximum allowable size of the selected network 206(1-3). In block 706, messages sent to the messaging API 512 from the application program 406b are tested to determine whether they exceed the maximum allowable size for the selected network 206(1-3). If the message size does not exceed this limit, the messages are dispatched according to the flow described in Figure 6 as shown in block 708.

[0129] If one or more of the messages are smaller than the limit, then multiple messages can be packaged to reduce overhead for a number of messages. This may be accomplished by placing a number of smaller messages into a packet for transmission over the selected network 206(1-3). This reduces the cost and latency for sending messages over networks that have an overhead cost or additional latency cost for each packet.

[0130] If, on the other hand, the size of one or more of the mes-

sages exceeds the maximum allowable limit, then the oversized messages may be disassembled as shown in block 710. To carry out block 710, the oversized messages may be compared to a prescribed message size to determine the more optimum method for disassembly. The prescribed message size may be used to the balance between efficiencies of sending larger chunks and disassembling the message into smaller chunks, as described previously. This prescribed message size may be an arbitrary or learned byte size specified by the system or it may be a maximum allowable limit dictated by the network on which the message is to be transmitted. Oversized messages that exceed the prescribed message size may be disassembled using the larger or smaller chunk size as shown in blocks 712, 714.

[0131] In block 716, the oversized messages are disassembled according to the determination of which chunk size to use. Disassembly may be carried out using identification coding or otherwise marking the order in which the portions of the disassembled messages should be reassembled at the OBU 105. The disassembled messages may then be transmitted to the OBU 105 as shown in block 718.

[0132] The multi-part message manager 526 keeps track of the portions of dissembled messages that have been transmitted and the portions that have not been transmitted. This allows only the un-transmitted portions of the messages can be later transmitted in case of a failure of system 100 or an out-of-range fault during message transmission. Accordingly, when the system 100 comes back on-line or the OBU 105 comes re-enters the coverage area of one of the wireless communication networks 206(1-3), then entire messages not need to be re-transmitted even if the messages are subsequently routed onto a different one of the wireless communication networks 206(1-3).

[0133] In block 720, the messages along with re-assembly information are received in the multi-part message manager 526 of the OBU 105. Like the multi-part message manager 526 in the system server 202, the multi-part message manager 526 also maintains a record of the portions of messages that have been received in case of a failure of the system 100 or an out-of-range fault. Once re-assembled, the messages are sent to the application program 410a of the OBU 105 as shown in block 722. As noted, the above-described communication are for exem-

plary purposes only and carried out by in the reverse direction and/or with devices other than the server-system 202 and OBU 105.

[0134] Figure 8 illustrates a flow chart depicting another communication flow 800 between the service server 202 and the OBU 105. In communication flow 800, one or more messages are dispatched from the server system 202 to a remote unit, such as OBU 105. Like above, the flow 800 may also be carried out by in the reverse direction, i.e., originating from the OBU 105 and terminating at the server system 202. Moreover, other devices besides the server system 202 and the OBU 105 may carry out the following flow.

[0135] At block 802, the server-system portion of the WCF architecture receives one or more messages from one or more of the application programs 410b. Before sending the messages, however, the application programs 410b may assign a particular priority to each of the messages. This priority, for example, may be set to a low priority, which indicates that the message or messages need not be sent with urgency. Alternatively, the priority may be set to a high priority, which indicates that the message or messages need to be sent with some degree of urgency and

be delivered soon.

[0136] The priority may also be set to a batch priority. The batch priority may indicate to the WCF architecture that the messages may be held in a queue, e.g., the message store 520, until other messages with the same priority level are accumulated. Once accumulated, the group of messages can be then sent as one batch.

[0137] At block 804, the API 512 may determine which of wireless communication networks 206(1-3) are available to the OBU 105. Each of the messages" priority is reviewed to determine whether high, low, multi-formatted (mixed processing) or batch priority conditions exist, as shown in block 806.

[0138] High priority processing is carried out for messages having a high priority, as shown in block 808. Low priority processing is executed for messages having a low priority, as shown in block 810. Batch priority processing is executed for messages having a batch priority as shown in block 812. In block 814, multi-formatted or mix processing may be carried out if the message priority is assigned by the application programs 410b. As noted, the above-described communication flow 800 is for exemplary purposes only and carried out by in the reverse direction

and/or with devices other than the server-system 202 and OBU 105.

3. SYSTEM OPERATION EXAMPLES

[0139] The overall system 100 can support many possible services and applications, examples of which are described below and illustrated in Figures 9 through 13. As noted above, the system 100 shown in Figures 1 and 2 illustrate one possible relationship between services and applications for a system 100 using an ASP-based model. In one embodiment, a group of core services 350 that perform vehicle-specific operations are available to the applications 108, 110. As noted above, the applications 108, 110 allow a user to customize the interpretation and display of the vehicle-specific operations based on the user's own requirements. The core services 350 act as building blocks of services that can be selected or combined in any desired manner, and can be accessed by or with any applications 108, 110 in the system 100; in other words, the applications 108, 110 act as a functional layer over the more primitive core services 350. For example, the core services 350 may be accessed by a help desk application to obtain vehicle location along with parametric data or by a service application to obtain parametric data and to per-

form functional tests. Because the system 100 can leverage other applications and services that the system 100 itself may not have and couple them with its own applications and services, the system 100 provides a flexible and adaptable platform that can accommodate many different needs.

[0140] In one embodiment, the applications may assemble the core services to perform specific functions. For example, one of the core services 350 may capture measured values and/or change parameters or operational settings in the vehicle 104 while the applications 108, 110 organize and process information from the core services 350 into groupings that are meaningful to a given user. A service outlet, for example, may want different data and/or data organized in a different manner than a leasing organization or a component manufacturer.

[0141] As noted above and as shown in Figures 1 and 2, the interface 106 can be a browser interface or graphical user interface (GUI) that allows a human user to access the system 100, or a machine-to-machine application programming interface (API). The user interface 106 allows the system 100 to act as a gateway between the user and the vehicle(s) 104 via the applications and services.

[0142] As noted above, the core services 350 provided by the system 100 acts as building blocks that can be assembled by applications in a variety of ways that can best serve the user. Possible core services 350 include:

[0143] *Parameters:* obtains discrete or data stream-based vehicle parameters, including standard and proprietary messages, upon user request;

[0144] *Alerts:* notification of the occurrence of a particular event (e.g., receipt of a trouble code or a notification of a parameter value occurring outside an acceptable parameter range);

[0145] *Functions:* runs functional tests on vehicle components and generates result reports;

[0146] *Configuration:* performs remote configuration of a vehicle or component by, for example, changing one or more vehicle parameters;

[0147] *Reprogramming:* performs complete reprogramming, or "re-flashing" of a selected on-vehicle controller;

[0148] *Geographic location:* provides vehicle location through, for example, a GPS system.

[0149] The list of core services 350 above is not meant to be exhaustive, but are simply examples of possible services that can be available directly to users or supplied to appli-

cations for further processing. Note that although the explanations below focus on obtaining data from a vehicle ECU 308, the system and functions described below are applicable for any vehicle data.

[0150] The "Parameters" service may include a simple parameter retrieval service as well as more sophisticated parameter retrieval services that address limitations in obtaining vehicle data when, for example, the vehicle is turned off. Figure 9 illustrates one simple process 900 for obtaining a parameter. When the OBU 105 receives a command from the system server 202 to retrieve a data value at block 902, the OBU 105 sends a query message to the ECU 308 to obtain the ECU's current reading at block 904. Once the ECU 308 returns a parameter value at block 906, the OBU 105 retrieves the value and forwards it to the system server 202 at block 908. Note that frequently used parameters may be packaged and transmitted to the system server 202 as a single message as a more efficient way of transferring data. Further, the specific means for getting a particular data item may depend on the specific requirements of a given ECU 308. For example, as is known in the art, data points corresponding to an anti-lock brake system may be obtained in a different manner than data

corresponding to engine coolant temperature.

[0151] Figure 10 is a flowchart illustrating one possible process to be offered as a "Parameters" service that is more sophisticated that the simple parameter retrieval service explained above. Although parameter data can simply be read from the vehicle's electronic controllers and provided to the user on demand, the "Parameters" service can also provide more sophisticated parameter data capture methods such as the type shown in Figure *10.* Figure 10 illustrates a "snapshot" process 1000 for obtaining a set of parameter values over a period of time, where the reporting of the parameter values is triggered by a specified event. Offering this service as an on-vehicle diagnostic tool is particularly helpful for intermittent fault diagnosis and vehicle performance analysis. Further, gathering data sets at prescribed periodic intervals minimizes negative effects caused by inherent problems in wireless communication systems, such communications drop-out and lack of coverage, which would normally make remote diagnostics ineffective.

[0152] To carry out the snapshot process 1000, the system 100 first initializes at block 1002 by, for example, storing the diagnostic parameters to monitor, setting the time inter-

vals at which parameter values are captured, selecting the number of captured values to be included in a single report, and selecting the event that will trigger reporting of the captured values. This information can be inputted into the OBU 105 via the interface 106. The parameter values to be captured can be any parameters accessible on the vehicle's electronic controllers by means of a diagnostic data stream or from discrete inputs on the OBU 105. The triggering event can be any non-continuous event that is monitored on the vehicle, such as the capture of an active trouble code from a vehicle controller or a parameter moving outside an established acceptable range.

[0153] Once the OBU 105 has been configured (block 1002), the OBU 105 maintains a number of historical value sets at block 1004 by caching a selected number of parameter readings during normal vehicle operation. While the OBU 105 captures the parameter readings, it also waits for the triggering event to occur. Once the trigger event occurs (block 1006), the OBU 105 continues caching the configured parameter readings occurring after the event (block 1008). The number of historical value sets can be, for example, half the number of captures to be included in the final report, while the number of value sets taken after the

triggering event can make up the other half. Note that the OBU 105 may, in another embodiment, capture parameter readings only before or after the triggering event as well or capture different numbers of values on either side of the event.

[0154] After all of the desired value sets have been captured and collected, all of the captured readings, both before and after the event, are combined into a final report at block 1010. The report can be stored on the OBU 105 for later retrieval or sent via wireless connection to the application server 202a for immediate viewing.

[0155] Another possible process that can be offered as a "Parameters" service is a "get stored values"(GSV) process 1100, as illustrated in Figure 11, for collecting parameter values from a vehicle even if the vehicle is unable to provide current parameter values at the time of the query. The GSV process 1100 addresses a situation where the vehicle controllers 308 are unable to respond to a query by the OBU 105 (e.g., while the vehicle is turned off) to respond to a query. This process is particularly useful in applications requiring remote retrieval of time-sensitive data, such as an odometer reading at the end of a scheduled period, or in any application where the vehicle operating

state is unknown at the time of the query.

[0156] For the GSV process 1100 to be effective, the OBU 105 should be designed to allow continuous remote access so that the OBU 105 is always ready for receiving and sending messages. The OBU 105 is initialized by receiving an instruction to periodically collect specified parameter data at a selected query time interval (block 1102). After receiving this command, the OBU 105 will periodically collect data at the specified query time intervals (block 1104). The values gathered by the OBU 105 are stored in the on-board unit's memory, such as a flash memory, at block 1106 before the OBU 105 is shut down when the vehicle 104 is turned off.

[0157] If the OBU 105 receives a GSV "retrieve" command from the application server 202a (block 1108), the OBU 105 checks the state of the vehicle controller 308 at block 1110. If the vehicle controller 308 is accessible, then the OBU 105 collects the current values from the vehicle controller 308 at that time and sends the collected current values to the system server 202 (block 1112). If the vehicle controller 308 is not available at the time of the command (e.g., if the vehicle is turned off), making the current values of the controller 308 irretrievable, the saved

values in the OBU 105 are sent back to the system server 202 as the retrieved values (block 1114).

[0158] By periodically collecting data at selected intervals while the vehicle controller is operational, the OBU 105 can at least obtain recent vehicle controller parameter readings before the controller 308 is inaccessible at some later time. As a result, the GSV process 110 allows the system server 202 to obtain the most recent controller data if current data is not available at the time of the query. In short, the GSV process 1100 returns the last known value in memory to the system server 202 if the vehicle is turned on and will retrieve a backup value, which may still be the last known value in memory, if the vehicle is turned off.

[0159] Multiple "Alerts "services may also be provided as a core service in the system 100. In its simplest form, the "Alert "service monitors vehicle trouble codes and transmits a message to the OBU 105 when the trouble code occurs. For example, a fault code may come as solicited or unso-licited, depending on how the controller 308 in the OBU 105 is instructed to handle faults. To obtain an unsolicited fault, the OBU 105 monitors the vehicle data bus 307 for the occurrence of a fault and notifies the system server

202 if a fault occurs. If only a set of individual faults are monitored, additional parsing shall be performed to filter out unwanted faults. For example, if a user only wishes to be informed of fault codes corresponding to a component breakdown, as opposed to being informed of all fault codes, the user can indicate this preference via the interface 106.

[0160] To obtain a solicited fault, the user may set up periodic queries to the OBU processor 300 in addition to setup notification. Note that the response message may match the message template even if no fault actually existed; in this case, additional parsing of the response message may be necessary to obtain useful information. For example, if the user solicits a request for information, the user may obtain a response based upon the criteria of that request, which may be different than the criteria for unsolicited notifications.

[0161] If desired, the "Alert" service may include additional functions such as providing the ability to add/remove individual faults, canceling the alert function for a given fault when a fault alert is fired so that only the first fault occurrence (and not subsequent fault occurrences) trigger a notification message, or configuring the "Alert "service to

be stored permanently in, for example, the database server 202d until the user removes the service or until the service is cancelled by a fault occurrence.

[0162] With respect to the example shown in Figure 12 and as noted above, the "Alerts" service may include among its functions the detection of a particular event by checking whether a monitored value exceeds a selected threshold. Note that although this example focuses on one diagnostic parameter, any number of diagnostic parameters may be combined into an algorithm to detect threshold limit boundaries. Further, values may be monitored over time, rather than as one single alert-triggering event, to monitor patterns and trends and to detect events more accurately.

[0163] As an example of an "Alert "service that monitors over time, Figure 12 shows an "Over RPM "threshold alert example that detects if a vehicle driver is abusing the vehicle engine. In this example, the Over RPM threshold alert considers the amount of time that the RPM level exceeds a specified limit (6000 RPM in this example) rather than simply generating an alert each time the RPM exceeds the level. The time delay ensures that alerts are generated only for events that may cause genuine concern.

[0164]   As shown in Figure 12, if the RPM exceeds 6000 RPM for a brief period that is less than 2 seconds (at 1202), the "Alert "service does not generate an alert. However, if the RPM does exceed 6000 RPM for more than two seconds (at 1204), the service fires an alert 1206 and resets itself (at 1208) when the RPM drops back below 6000 RPM. The actual circuitry for monitoring RPM and implementing this example of the "Alert" service in the system 100 (e.g., on the OBU 105) is well within the skill of those in the art. Further, the time delay concept shown in Figure 12 can be used for any parameter where undesirable operation is preferably detected via time and value thresholds.

[0165]   The "Alert "services may also include a tamper alert feature that, as shown in Figure 13, allows the user to monitor any unauthorized alteration of configurable parameters. This feature 1300 generally contains a setup process 1302 and a tamper check process 1304. When a user requests the tamper alert service (block 1306), the OBU 105 captures the value of the parameter at the time of the request and saves the parameter value to a file in the OBU's memory (e.g., OBU's flash memory 315 or DRAM 316) at block 1308. Note that this parameter retrieval process may involve using the "Parameters "service as explained

above to query the ECU or other vehicle controller or component 308.

[0166] The actual tamper check process is conducted when, for example, the vehicle is initially turned on. At this point, the OBU 105 checks the parameter again to get its current value at the time the vehicle ignition is turned on (block 810). If the current value is different than the saved value (block 1312), a tamper alert message will be returned to the user (block 1314). If the compared values are the same at block 1312, however, the vehicle continues operation as usual without transmitting any tamper alert signal (block 1316). In one embodiment, the user may add/remove individual tamper alerts, and the tamper alert may be cancelled at block 1318 once the alert is fired.

[0167] A "Change Parameters "function may also be included as part of a configuration core service, as shown in Figure 14. This feature may allow a user to remotely insert or update, for example, a parameter or message definition in the vehicle. As shown in Figure 14, the function 1400 includes receiving a parameter change request (block 1402), receiving the specific parameter to be changed in the vehicle (block 1404), changing the parameter (block 1406), and saving the parameter in memory (block 1408). In one

embodiment, the updated parameter definitions are stored permanently in memory until the next change request. Further, in one embodiment, the updated definition takes effect as soon as the update is completed.

[0168] The core services can be accessed by one or more applications, as noted above. The system 100 may include the ability to leverage other services that it may or may not have, such as, Fuel Tax Reporting/State Line Crossing applications, Asset tracking/utilization programs, Driver Performance applications, On-line Vehicle Documentation, detailed mapping applications, etc. This flexibility, coupled with modular services and applications 108, 110 that can be added, subtracted, and combined at will, provides for a very flexible and adaptable platform.

4 APPLICATIONS

[0169] As described above, the system 100 allows users to customize their own vehicle monitoring, programming and diagnostics systems based on their own specialized needs by offering a plurality of applications that can be selected and combined in a modular fashion as desired by the user. The applications may include service offerings such as Remote Diagnostics, Fuel Economy, Trip Reporting, Automatic Vehicle Location based upon GPS or satellite

based system information, and others. The applications listed here and described in greater detail below are only examples and are not meant to be limiting or comprehensive in any manner. Those of skill in the art will understand that other applications may also be included as possible application options.

[0170] A "Remote Diagnostics application for example, provides the ability to perform component analysis before or during a vehicle breakdown and allows vehicle maintenance locations to receive parametric information from a vehicle prior to its arrival, or prior to dispatching a technician to the vehicle. Further, the "Remote Diagnostics" application allows a technician to perform selected diagnostic tests on the vehicle or system, with the test process being managed by the OBU 105. In one embodiment, the "Remote Diagnostics" application allows a user to view parameters, active and inactive fault codes, and vehicle configurations, for example, and may also allow authorized users to perform functional tests and configuration changes on the vehicle. Remote Diagnostics may be initiated when, for example, a vehicle notifies a fleet manager about a series of established alerts or when the diagnostics are requested manually by a fleet authorized user. In practice,

the application may provide diagnostic applications via the inventive system 100. When the user logs on to the system 100 via the interface 106, for example, he or she may be presented with a list of vehicles that have reported alerts or notifications that may need attention. If no alerts are active, the user is provided a list of vehicles for which he or she is responsible. At this point the user may elect to use a remote diagnostics application, such as the remote diagnostics application described below and shown in Figure 15, 1513, to perform further analysis on the vehicle to determine the severity of the problem.

[0171] Figure 15 is a block diagram illustrating one possible overall web site architecture 1500 that includes a remote diagnostics application. In this example, a user may log into the application (block 1502) to reach an application home page 1504. From the home page, the user may access a range of information, such as fuel economy 1506 or leased vehicle information 1508, or access utilities 1510 or a remote diagnostics application (RDA) page 1512 to monitor, diagnose, and/or reprogram vehicle parameters. In this example, the utilities 1510 allow the user to define and/or modify vehicle groups 1514, specific vehicles 1516, and alerts 1518. The RDA page 1512 pro-

vides users with access to, for example, vehicle information and parameters 1520, including pages that allowing parameter viewing 1522 and reprogramming 1524. Note that other architectures and implementations are possible for this application as well as other applications without departing from the scope of the invention.

[0172] A "Leased Vehicle Management"(LVM) application is another possible option to generate periodic status reports summarizing selected parameters for each vehicle in a fleet, such as total vehicle distance, total idle fuel, total idle time, total fuel used, and/or total fuel economy. Figure 16 is a block diagram illustrating one possible example architecture for the Leased Vehicle Management application 1600. In this example, the user reaches a main page 1602 that allows the user to choose between a group details page 1602 and the task details page 1606. Group details 1604 correspond to all information for a selected vehicle group, while task details 1606 correspond to all information for a selected task. The group details page 1604 may allow the user to, for example, create new tasks (e.g., the timing of data collection for a selected vehicle group) 1608, generate a report list 1610, or generate more detailed reports listing specifying, for example,

parameter values for a selected report 1612. The task details page 1606 includes similar options, allowing the user to view reports for a selected task 1614 and generate more detailed reports 1616. The task details page 1606 also allows a user to stop a task 1618 or delete a task 1620.

[0173] An "Engine Management" application may also be an option to target fleets whose vehicles encounter varying road and traffic conditions, and varying load types and weights. The objective of the "Engine Management" application is to improve overall fleet fuel economy via dynamic control of a vehicle"' operational characteristics, in particular, horsepower ratings and maximum road speed limits. Traditionally such operating parameters have been established once at a fleet wide level, not taking into consideration some of the variables listed above. In addition, making these changes required physical contact with the vehicle, necessitating undesirable vehicle downtime. Dynamic adjustments based upon operating conditions can provide reductions in vehicle operational costs, thus resulting in significant savings at a fleet level. With this application the user will be able to dynamically configure the vehicle wherever it may be; tailoring its operational characteristics

based upon route, load, and other vehicle operation factors. The "Engine Management" application may include both measured parameters and programmable parameters. Examples of programmable parameters include Vehicle Road Speed Limit, Engine Horsepower/Torque, Engine Idle Shutdown Time and Cruise Control Settings.

[0174] A "Trip Report" application may also be provided as an option. This application allows the fleet manager to obtain trip information from the vehicle on a near-real-time basis. The user can analyze trip information for single vehicles as well as any increment of their fleet. This application primarily uses measured parameters such as odometer readings, total trip fuel, idle fuel, average fuel economy, vehicle route taken, and others. It also uses some parameters to derive data, such as total idle hours and the type of idle hours recorded. The output from this application can also be used as input to the billing systems of leasing companies who charge customers based upon mileage.

[0175] A "Maintenance Alert" application allows the fleet manager to establish a series of maintenance triggers based upon key parameters. When a parameter threshold is encountered, the fleet manager will be notified automatically by

the system, thus initiating a maintenance event without physical contact with the vehicle. For example, a fleet may establish a preventive maintenance cycle based upon odometer reading. If the system server 202 is made aware of the interval, it can notify the fleet of the precise moment when that interval occurs. Alerts may provide notification on parameters such as diagnostic codes, fluid levels and parameter ranges as well as unauthorized tampering with the vehicle.

[0176] A "Vehicle Configuration" application may be offered to allow a fleet manager to set certain parameters for multiple vehicles in a fleet so that the selected vehicles will operate within its established standards. Examples of parameters include horsepower ratings, maximum road speed limits, maximum and minimum cruise control set speeds and maximum engine idle time before shutdown. Traditionally, this step has required a physical connection of a diagnostic application or tool to the vehicle, but physical connections are time-consuming and require the same step to be repeated on every vehicle that is serviced. The wireless nature of the "Vehicle Configuration" application allows operational settings and alerts for several vehicles within a fleet at one time by allowing the user to

identify selected vehicles, set parameters, and initiate an automated process where each vehicle is systematically configured with the same parameter settings.

[0177] A "WarrantyManagement" application may also be offered as part of a data mining strategy used by, for example, original equipment manufacturers (OEMs) to help diagnose warranty relationships between major components or to assess warranty claims for validity. This application would, for example, obtain detailed vehicle data from the database server 202d, using both fleet specific and system-wide data mining, and then correlate the data with warranty requirements.

[0178] As noted above, the possible modular applications described herein are meant as illustrative examples only. Further, as noted above, the applications 108, 110 accessed by the infrastructure 102 can be generated by third parties and offered as modules for incorporating into a particular user"' interface 106 and accessing the OBU 105 and other system-supported core services and applications. The modular functionality offered by independent applications 108, 110 allows disparate users to access the same vehicle data via the same OBUs 105 and the same infrastructure 102, but be offered customized data, func-

tionalities, and interfaces that are meaningful to that user'" industry as determined by the particular modular applications selected by the user. The specific manner for implementing the applications via, for example, computer programs, is within those of skill in the art.

[0179] The inventive system therefore provides a modular wireless vehicle diagnostics, command and control system that is customizable to a user'" specifications. More particularly, the modular applications 108, 110 provide much versatility and allow users from disparate industries to use the same overall inventive system 100 by selecting the applications 108, 110 relevant to their particular industry. Further, by creating a wireless diagnostics and command and control service, the invention provides real-time remote access to vehicles and vehicle systems via, for example, the Internet and wireless networks. In one embodiment, the inventive system allows users to connect to multiple data points on any given vehicle to interpret and analyze the vehicle data in real time, change vehicle parameters as needed and create historical databases to guide future decisions. Further, by monitoring vehicle operation in real time and providing customized reports for each vehicle, the inventive system achieves high operating

efficiency, lowered maintenance costs and downtime, and even allows pre-ordering of parts as vehicles approach scheduled maintenance.

[0180] Note that the capabilities described above are meant to be illustrative and not limiting. The system 100 can be adapted to, for example, establish a setting that is applied to selected group of vehicles with a single command rather than individually establishing a setting for each vehicle. The aspects of the request, including authorization, vehicle/component differences, password differences, and configuration limitations of the specific request, may be managed by, for example, the system server 202. In another embodiment, the system 100 can view each vehicle 104 as a single entity to allow the user to communicate with multiple ECU'" on the same vehicle 104 at the same time. For example, data can be obtained from an Engine ECU and Transmission ECU at the same time, with the resultant data from each controller correlated to the other to add more detail to the data offered to the user.

[0181] Variations of the system described above are possible without departing from the scope of the invention. For example, selected applications may be run locally on proprietary equipment owned by the customers (i.e., the fleet

managers, vehicle distributors, vehicle dealers and the like) as a stand alone software application instead of over the Internet. Further, the OBU 105 can be equipped with, for example, a bar code scanner and/or other human user interface to facilitate data capture. Other user interfaces and functions, such as a handheld diagnostics tool, workflow integration tool, links between data captured by different applications, and tools to provide advance warning of vehicle faults or pre-arrival diagnostics information may also be included as application modules or core services or even integrated within the application modules themselves. Note that one or more additional servers can also be incorporated into the system to, for example, accommodate additional data management functions and/or provide interfaces for integrating with existing applications.

[0182] Information obtained via the inventive system can also be used to, for example, re-calibrate vehicle components, perform firmware downloads, perform component failure analysis, determine wear characteristics, analyze quality of components used in their manufacturing processes, retrieve and manage warranty information, receive indications of vehicle maintenance needs, monitor vehicle use

and abuse, and/or monitor lessee trip information, perform proactive data analysis, perform pre-arrival diagnostics, re-calibrate vehicle components, and/or perform firmware downloads. Note that this list of options is not exhaustive and those of skill in the art will understand that other variations in the data obtained via the inventive system and how the data is presented and used can vary without departing from the scope of the invention.

[0183] It should be understood that various alternatives to the embodiments of the invention described herein may be employed in practicing the invention. It is intended that the following claims define the scope of the invention and that the method and apparatus within the scope of these claims and their equivalents be covered thereby.